Homework 4 **CSCE 421**

Due: 7.00pm on November 6, 2019

Instructions for homework submission

a) Please write a brief report for the experimental problems. At the end of the pdf file, please include your code. The code has to be directly converted instead of scanned (i.e. the text in the code must be selectable).

b) Staple all your answers and **hand them out in paper** in class or in the instructor's or TA's office.

c) Please start early :)

Question 1: News article popularity estimation

In this problem our goal is to estimate the popularity of online articles published over the last years. The provided data contain articles published by Mashable in a period of two years. The goal is to predict the number of shares in social networks, which approximates the popularity of an article. The dataset is taken by the UCI repository: `https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity#`. You can find the train and test data under *OnlineNewsPopularityTrain.csv* and *OnlineNewsPopularityTest.csv*, respectively, and the data description under *OnlineNewsPopularity.txt*, in "Homework 4" folder on Piazza. The goal is to predict the number of shares (last column of .csv file) based on all the other available features. **For questions (a)-(d) you can ignore the first feature ("url").**

(a) (3 points) *Decision tree regression:* Estimate the number of shares of an article on the test data using a decision tree regression. *Implement* a 5-fold cross-validation on the train set to determine the optimal depth of the decision tree. Report the average error (e.g., square root residual sum of squares) over all folds on the train data for each depth of the tree, as well as the average error of the test data for the optimal tree depth.

Note: You can use any library for the decision tree regression, but you have to implement the 5-fold cross-validation.

(b) (4 points) *Random forest regression: Implement* a regression tree to estimate the number of shares of an article on the test data. You can use any library for the decision tree, but you will need to implement the random forest (i.e., randomize the input samples and the input features of each decision tree, combine the results from all trees to obtain a final decision). *Implement* a 5-fold cross-validation on the train to identify the optimal three depth and number of trees. Report the average error (e.g., square root residual sum of squares) over all folds on the train data for each combination of tree depth and number of trees (you can show this as a 2-dimensional color-coded matrix, whose x/y dimensions are the number of trees and tree depth, and the color-coding reflects the average error over all folds). Report the average error of the test data using the random forest with the optimal tree depth and number of trees.

(c) (1 point) *Feature exploration:* Inspect the final decision tree from question (a) and identify the most important features for predicting the popularity score. Please report these features and provide your intuitions.

(d) (2 points) *Feedforward neural network:* Estimate the number of shares of an article on the

test data using a feedforward neural network. Using a 5-fold cross-validation experiment with different number of layers, node per layer, activation functions, dropout, and learning rate. Experiment with the original feature set, as well as with the most important features found from question (iii).
Hint: You can use the *sklearn.GridSearchCV* function or any other similar function for the hyper-parameter tuning.

(e) (1 point) *Bonus - NLP feature extraction:* Use the first column (name "url") as a feature to the regression. This feature does not contain numerical values, but includes the url of the corresponding article. Take the last part of the url (e.g., "amazon instant video browser" from "http://mashable.com/2013/01/07/amazon-instant-video-browser/") and extract natural language processing (NLP) features from this part using the *word2vec* library. Use these and your favorite regression model to estimate the popularity of the articles in the test set. Combine these features with the most important features from question (iii) and report the corresponding results on the test set. You will have to re-tune the hyper-parameters of your regression model to the new feature set.
Hint: *word2vec* converts an input text into numerical feature values using a bag-of-word approach. You can find more information on *word2vec* library under this link `https://code.google.com/archive/p/word2vec/`.